Session 207

USB In-Depth

Craig Keithley
USB & FireWire Technology Manager
Apple Worldwide Developer Relations

Session 207

USB In-Depth

Fernando Urbina
USB Software Team

What You'll Learn

- Overview of IOUSBFamily objects
- Writing Kernel and user space drivers
- How USB driver matching works
- How to debug USB drivers

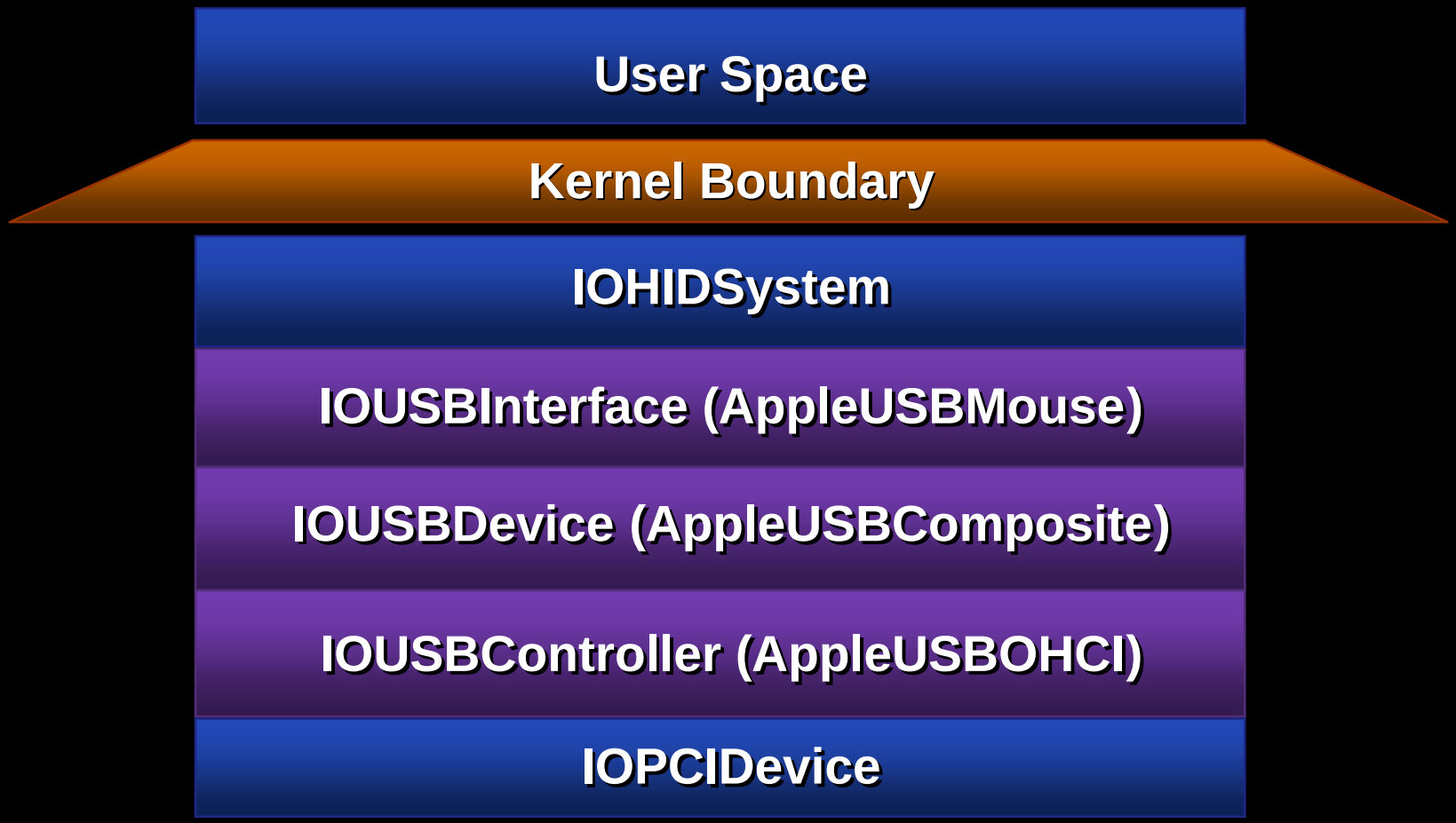
Mac OS X USB Overview

- Project Builder used to write USB drivers
- User space or Kernel space ?
- Kernel drivers use I/O Kit's object oriented features
- User space drivers make use of a IOKitUserLib "DeviceInterface"

Writing USB Drivers for Mac OS X

- USB Family objects are in the Kernel
- Drivers can access the objects from either Kernel space or User space

USB in Kernel Space



Kernel Mode USB Drivers

- Required for some types of USB devices
  - HID (mouse, keyboard)
  - Mass storage
  - Networking
- Apple recommends writing user mode drivers if at all possible!

Kernel Mode I/O Kit Drivers

- Written in subset of C++
- Base class is IOService
- Provide a matching dictionary for dynamic loading
- kext package in /System/Library/Extensions

IOUSBFamily Kernel Classes

IOUSBPipe

IOUSBPipe

IOUSBInterface

IOUSBInterfaceUserClient

IOUSBDevice

IOUSBDeviceUserClient

IOUSBFamily Kernel Classes

- IOUSBDevice
  - One per USB device connected to the bus
  - Provider is IOUSBController
  - Abstraction of a USB device
  - Instantiates and is the provider for IOUSBInterface

IOUSBFamily Kernel Classes

- IOUSBInterface
  - One per interface
  - Created by SetConfiguration
  - Provider is IOUSBDevice
  - Abstraction of a USB interface
  - Instantiates IOUSBPipe objects

IOUSBFamily Kernel Classes

- IOUSBPipe
  - One for the default control endpoint
  - One for each additional endpoint in each IOUSBInterface object
  - Methods for data flow control

IOUSBFamily Kernel Classes

- IOUSBDeviceUserClient
- IOUSBInterfaceUserClient
  - Provide connection between user tasks and kernel objects
  - Appears in registry when a user land task is connected to an IOUSBDevice or an IOUSBInterface

I/O Kit Convention for Apple Supplied Drivers

- IOxxx
  - May be subclassed
- Applexxx
  - Not intended to be subclassed

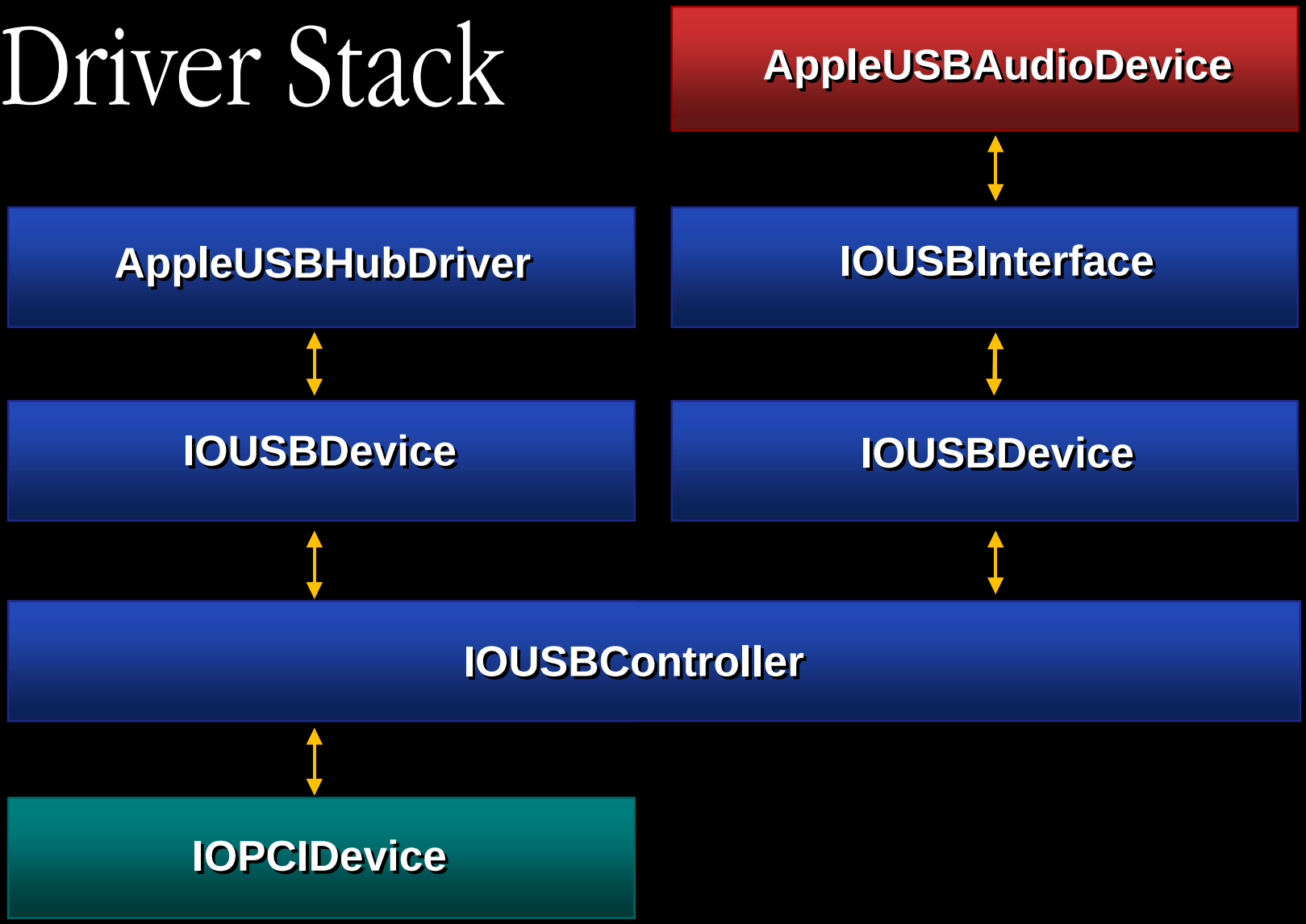
Apple USB Kernel Drivers

- AppleUSBMouse
- AppleUSBKeyboard
- AppleUSBComposite
- AppleUSBCDCDriver
- AppleUSBAudio
- AppleUSBHubDriver
- AppleUSBProKeyboard

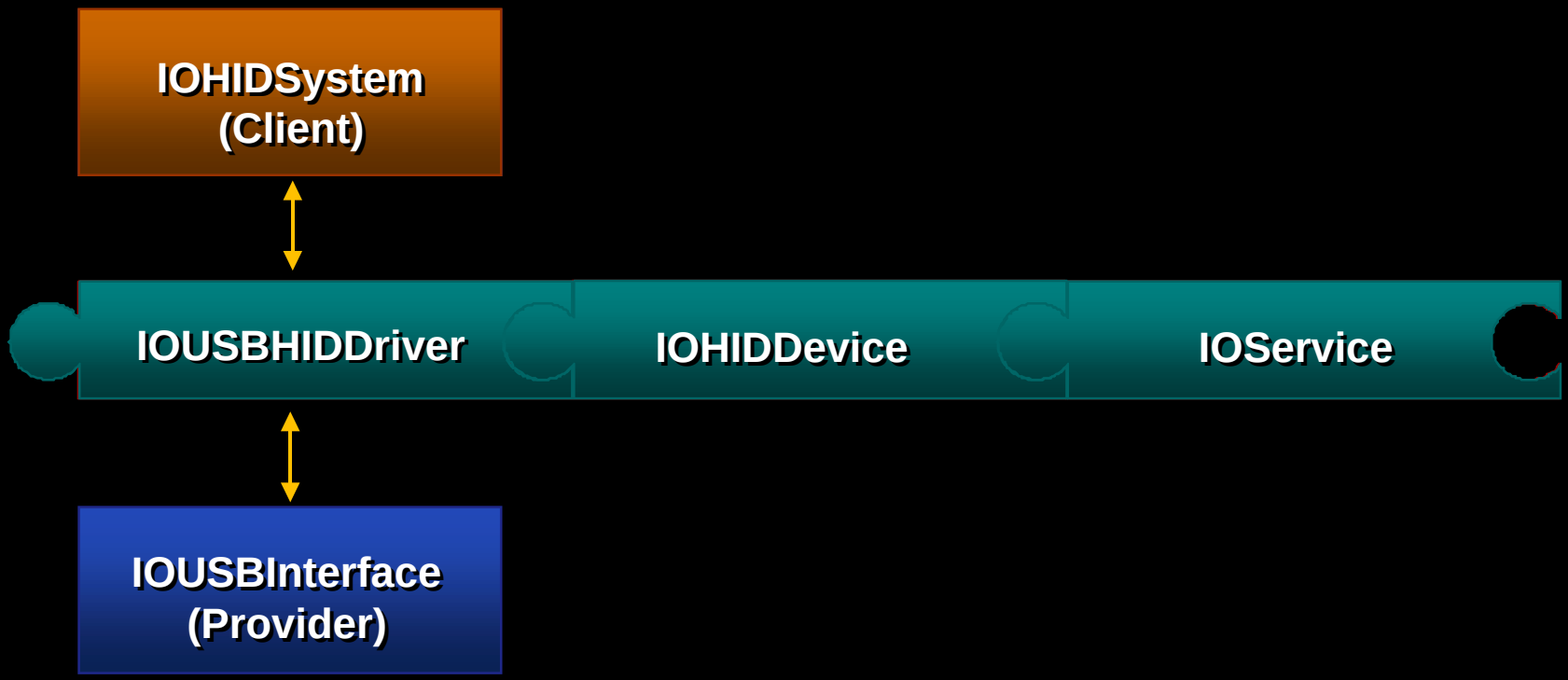
Subclassable USB Kernel Drivers

- IOUSBHIDDriver
- IOUSBMassStorageClass

I/O Kit Kernel Driver Stack



I/O Kit Driver Inheritance



Driver Example Code

- IOUSBFamily and most drivers are open-sourced
- Module name is IOUSBFamily
- **cvs.opensource.apple.com:/cvs/Darwin**

IOUSBFamily Driver Matching

- IOProviderClass (IOUSBDevice or IOUSBInterface)
- Family specific matching
  - Uses USB common class spec 1.0

USB Common Class Specification Device Driver Ranking

idVendor & idProduct & bcdDevice

idVendor & idProduct

idVendor & bDeviceSubClass & bDeviceProtocol

idVendor & bDeviceSubClass

bDeviceClass & bDeviceSubClass & bDeviceProtocol

bDeviceClass & bDeviceSubClass

bDeviceClass = 0xff

bDeviceClass ≠ 0xff

Active Driver Matching

- Driver may decline device based on probe
- Ranked list created and first one to successfully start wins!

Specifying Matching Dictionary

- Bundle settings tab of Project Builder
- IOKitPersonalities
  - CFBundleIdentifier
  - IOClass
  - IOProviderClass (IOUSBDevice or IOUSBInterface)
  - Matching criteria from USB CC spec section 3.10
  - IOKitDebug for debugging

Specifying Matching Dictionary

- OSBundleLibraries
  - **com.apple.iokit.IOUSBFamily “1.8”**
- OSBundleRequired “Root” might be needed

IOService Startup Sequence

- Driver matching
  - **init**
  - **attach**
  - **probe**
  - **detach**
- Driver startup
  - **attach**
  - **start**

IOService Termination Sequence

- message
  - kIOMessageServiceIsTerminated
- terminate
- stop
- detach
- finalize
- free

Session 207

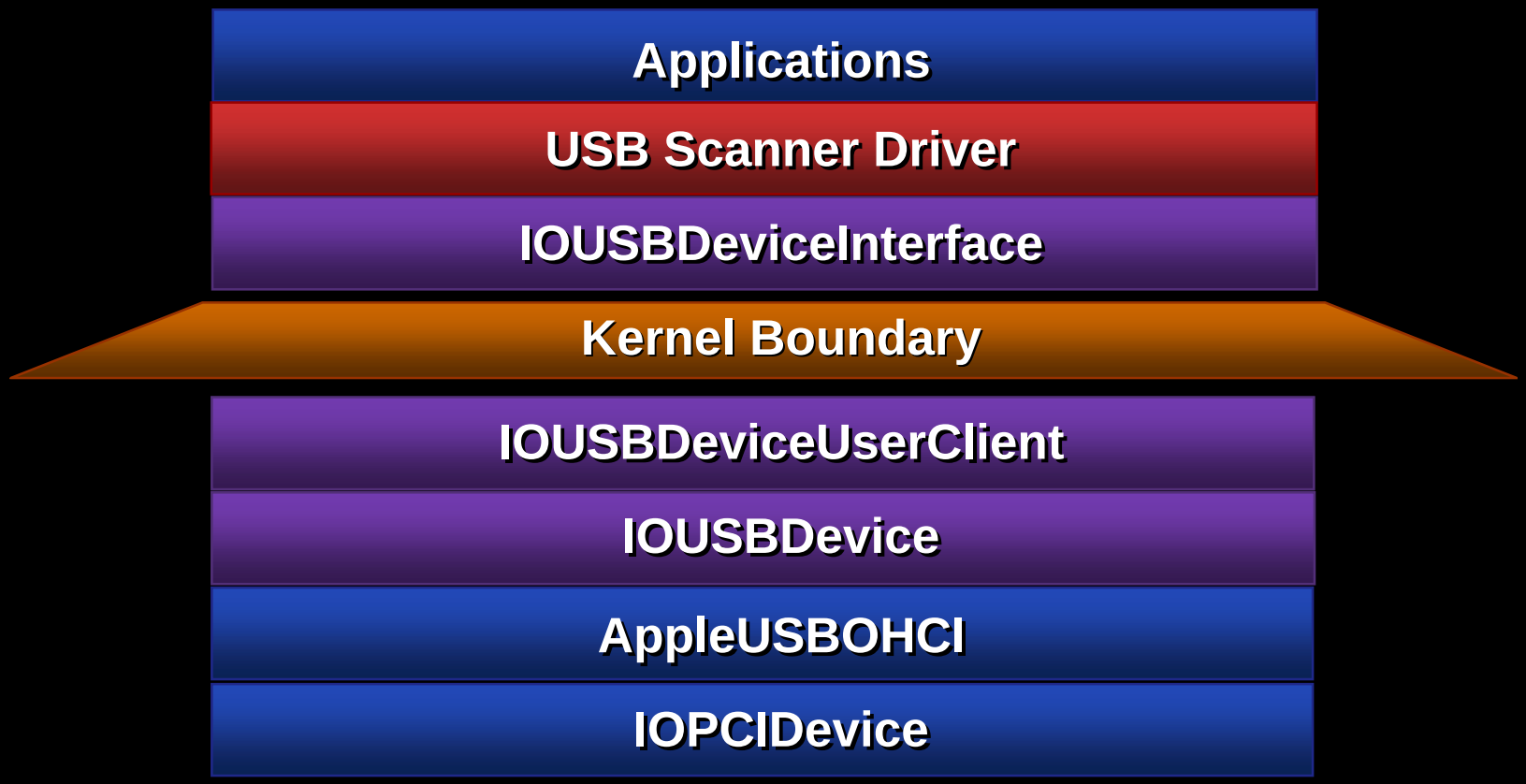
USB In-Depth

Rhoads Hollowell
USB Software Team

IOUSBFamily Workloop

- One per IOUSBController
- Used to serialize access to hardware
- Method **getWorkloop()** gets USB controller workloop
- No synchronous USB calls when operating inside the USB workloop
  - Can use **thread_call** mechanism

USB in the User Space



Writing a User Mode Driver

- Framework requirements
  - CoreFoundation.framework
  - IOKit.framework
- Uses CFPlugin model
  - IOUSBDeviceInterface plugin
  - IOUSBInterfaceInterface plugin

Writing a User Mode Driver

- IOKit API in **<IOKit/IOKitLib.h>**
- Plugin API in **<IOKit/IOCFPlugIn.h>**
- USB API in **<IOKit/usb/IOUSBLib.h>**
  - IOUSBDeviceInterface
  - IOUSBInterfaceInterface

Finding the Correct Kernel Object (io_service_t)

- Obtain master port for kernel communication
- Create matching dictionary
- Add qualifications to the matching dictionary
- Obtain kernel iterator (io_iterator_t) to matching services

1. Obtain Master Port

- **#include <mach/mach.h>**
- **err = IOMasterPort(MACH_PORT_NULL, &masterPort);**
- Should be released with **mach_port_deallocate(...)**

2. Create Matching Dictionary

- For IOUSBDevice objects:
  matchingDictionary = IOServiceMatching(kIOUSBDeviceClassName)

- For IOUSBInterface objects:
  matchingDictionary = IOServiceMatching(kIOUSBInterfaceClassName)

3. Add Qualifications to the Matching Dictionary

- See the USB common class specification, section 3.10
- In Mac OS X 10.0.3, USB matching is very strict!
- Be aware of different matching rules for IOUSBDevice and IOUSBInterface

4. Obtain the Kernel Iterator

- **err = IOServiceGetMatchingServices(**
  masterPort, matchingDictionary, &iterator);

- **while(usbDeviceRef = IOIteratorNext(iterator))**
- Iterator should be released when done:
  IOObjectRelease(iterator);

Using the io_service_t

- IOCreatePlugInInterfaceForService(
  usbDeviceRef,
  kIOUSBDeviceUserClientTypeID,
  kIOCFPlugInInterfaceID,
  &iodev,
  &score);
- IOCreatePlugInInterfaceForService(
  usbInterfaceRef,
  kIOUSBInterfaceUserClientTypeID,
  …

Using the IOCFPlugInInterface

- ```
  (*iodev)->QueryInterface(
                iodev,
                CFUUIDGetUUIDBytes
                        (kIOUSBDeviceInterfaceID),
                &dev);
  ```
- ```
  (*iodev)->QueryInterface(
                iodev,
                CFUUIDGetUUIDBytes
                        (kIOUSBInterfaceInterfaceID),
                &dev);
  ```
- ```
  (*iodev)->Release(iodev);
  ```
  ```

Using the IOUSBDeviceInterface

```
(*dev)->USBDeviceOpen(dev)

(*dev)->GetConfigurationDescriptorPtr(dev…)

(*dev)->SetConfiguration(dev, …)

(*dev)->CreateInterfaceIterator(dev, …)

(*dev)->USBDeviceClose(dev)

(*dev)->Release(dev)
```

Using the IOUSBInterfaceInterface

(*intf)->USBInterfaceOpen(intf)

(*intf)->GetNumEndpoints(intf, …)

(*intf)->SetAlternateInterface(intf, …)

(*intf)->GetPipeProperties(intf, …)

(*intf)->ReadPipe(intf, …)

(*intf)->WritePipe(intf, …)

(*intf)->USBInterfaceClose(intf)

(*intf)->Release(intf)

Asynchronous I/O

```
(*intf)->CreateInterfaceAsyncEventSource(…)
CFRunLoopAddSource(…)
(*intf)->ReadPipeAsync(intf, …)
CFRunLoopRun();
```

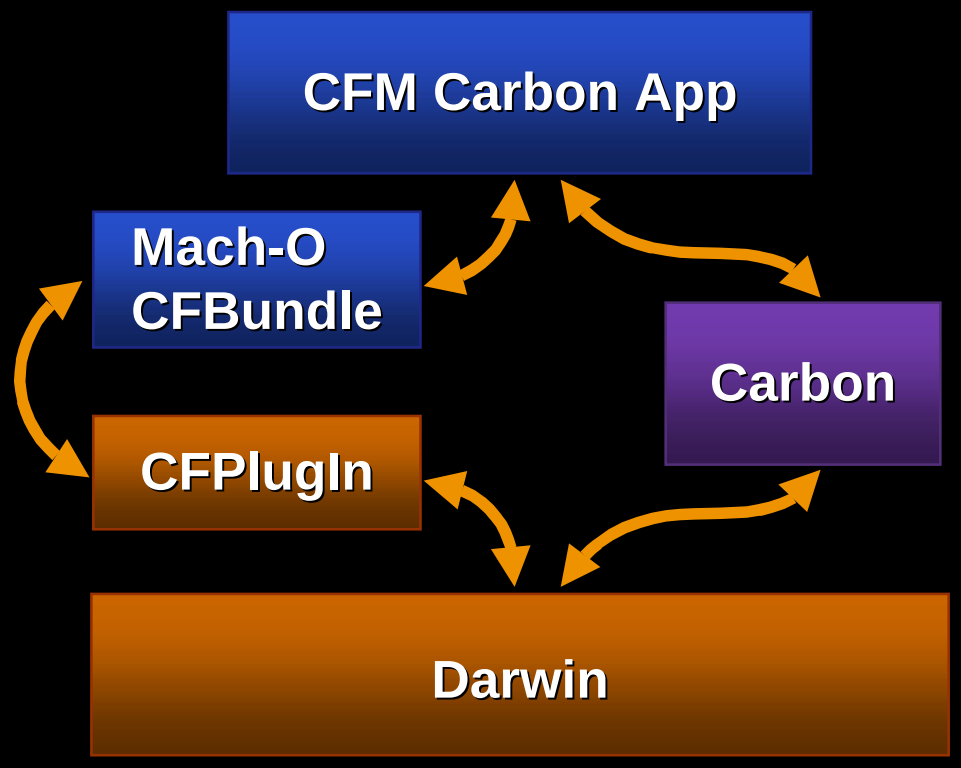
- See **<CFRunLoop.h>**

<IOKit/usb/IOUSBLib.h>

- In the IOKit.framework
- Full API for:
  - IOUSBDeviceInterface
  - IOUSBInterfaceInterface

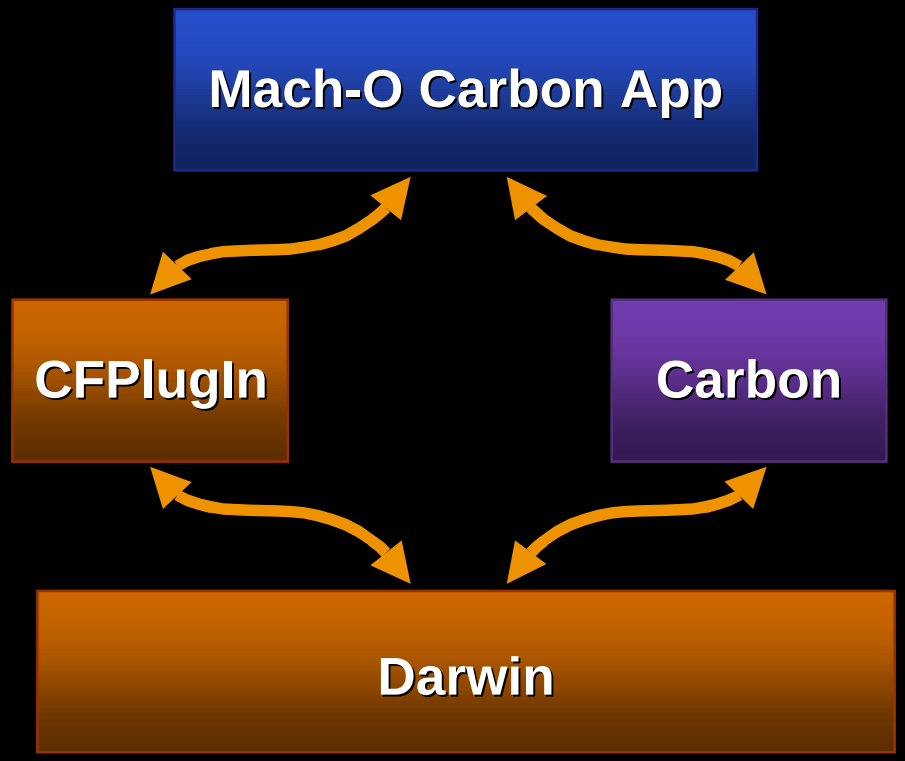
Access from CFM Carbon Apps

I/O Kit Native APIs available via CFPlugin



Access from Mach-O Carbon Apps

I/O Kit Native APIs directly available



Demo

Debugging

Rhoads Hollowell
USB Software Team

Resources

USB Implementers Forum
http://www.usb.org

Mac OS X Developer Information
http://developer.apple.com/macosx/

Macintosh USB Development Information
http://developer.apple.com/hardware/usb/

Related Sessions

104 I/O Kit Update: An overview of I/O Kit in Mac OS X	Room A1 **Mon., 3:30pm**
204 I/O Kit: Storage Drivers: Mass storage drivers in Mac OS X	Room A2 **Tues., 5:00pm**
117 Image Capture Framework: Image Capture Framework	Room C **Wed., 10:30am**
203 USB Overview: Overview of USB in Mac OS X	Room A2 **Tues., 3:30pm**
FF108 USB/FireWire Feedback Forum: Tell us what you think	Room J1 **Thurs., 2:00pm**

Who to Contact

Apple Worldwide Developer Relations

Craig Keithley
USB & FireWire Technology Manager
keithley@apple.com

USB Developer Mailing list

usb@isg.apple.com

http://developer.apple.com/events/urls.html

Q&A

Craig Keithley
USB & FireWire Technology Manager
keithley@apple.com

http://developer.apple.com/events/urls.html

WWDC

Worldwide Developers Conference 2001